S&H Form: PTO/SB/05 (12/97)

UTILITY PATENT APPLICATION TRANSMITTAL

(Only for new nonprovisional applications under 37 CFR 1.53(b))

Attorney Docket No. 21.1977

First Named Inventor or Application Identifier:

Jyunichi KAMAKURA et al.

Express Mail Label No.

APPLICATION ELEMENTS

See MPEP chapter 600 concerning utility patent application contents.

ADDRESS TO: **Assistant Commissioner for Patents**
Box Patent Application
Washington, DC 20231

1. [X] Fee Transmittal Form
2. [X] Specification, Claims & Abstract ..... [Total Pages: 22]
3. [X] Drawing(s) *(35 USC 113)* .... [Total Sheets: 11]
4. [X] Oath or Declaration .... [Total Pages: 3]
   a. [X] Newly executed (original or copy)
   b. [ ] Copy from a prior application (37 CFR 1.63(d)) *(for continuation/divisional with Box 17 completed)*
      i. [ ] DELETION OF INVENTOR(S)
      Signed statement attached deleting inventor(s) named in the prior application, see 37 CFR 1.63(d)(2) and 1.33(b).
5. [ ] Incorporation by Reference (usable if Box 4b is checked)
   The entire disclosure of the prior application, from which a copy of the oath or declaration is supplied under Box 4b, is considered as being part of the disclosure of the accompanying application and is hereby incorporated by reference therein.
6. [ ] Microfiche Computer Program *(Appendix)*
7. [ ] Nucleotide and/or Amino Acid Sequence Submission *(if applicable, all necessary)*
   a. [ ] Computer Readable Copy
   b. [ ] Paper Copy (identical to computer copy)
   c. [ ] Statement verifying identity of above copies

ACCOMPANYING APPLICATION PARTS

8. [X] Assignment Papers (cover sheet & document(s))
9. [ ] 37 CFR 3.73(b) Statement *(when there is an assignee)* [ ] Power of Attorney
10. [ ] English Translation Document *(if applicable)*
11. [ ] Information Disclosure Statement (IDS)/PTO-1449[ ] Copies of IDS Citations
12. [ ] Preliminary Amendment
13. [X] Return Receipt Postcard (MPEP 503) *(Should be specifically itemized)*
14. [ ] Small Entity Statement(s) [ ] Statement filed in prior application, status still proper and desired.
15. [X] Certified Copy of Priority Document(s) *(if foreign priority is claimed)*
16. [ ] Other:

17. If a CONTINUING APPLICATION, *check appropriate box and supply the requisite information:*

[ ] Continuation [ ] Divisional [ ] Continuation-in-part (CIP) of prior application No: ___/_______

18. CORRESPONDENCE ADDRESS

STAAS & HALSEY LLP
Attn: H. J. Staas
700 Eleventh Street, N.W., Suite 500
Washington, DC 20001

Telephone: (202) 434-1500
Facsimile: (202) 434-1501

System Analysis Apparatus and Method, and Storage Medium Storing System Analysis Program

TECHNICAL FIELD OF THE INVENTION

The present invention relates to system analysis technology, more particularly to system analysis technology used in order to divide an existing system into reusable subsystems and other resources, for example.

BACKGROUND OF THE INVENTION

Even if an existing system is reorganized, main functions in the new system are not very different from those of the existing system. In this situation, subsystems of the existing system are reused to reduce both of the development costs and development term. In order to reuse subsystems, it is necessary to isolate subsystems required for reuse from the other resources in an enormous existing system. Currently, this isolation of subsystems is carried out using design documents at the time of development and/or the knowledge of development engineers, and highly depends on the know how at the time of design. But, such a technique that depends on the human experience cannot definitely identify system resources to be reused, and it sometimes causes additional developments costs and period. There is a very real danger of cost increases during development due to design errors of the interface between the reused portions and the newly developed portions in addition to the isolation difficulty.

Also, if there are deficiencies in the design documents, and/or mismatches between the design documents and the current system, and/or the developer of the current system is absent, etc., it is not possible to smoothly investigate grouping of the subsystems in the existing system, reuse of the subsystems, new development, and/or application of packaged programs etc. Therefore, the system itself must be analyzed.

Further, now that almost all procedures are integrated into a system, the procedures themselves are incorporated into a black box and relationships between procedures become complicated which means that new development of all procedures is extremely difficult. In this respect a process for analyzing the existing system itself is absolutely essential in reorganization of the existing system.

SUMMARY OF THE INVENTION

The object of the present invention is therefore to provide a system analysis apparatus, method and program for analyzing an existing system.

It is also an object of the present invention to obtain information to be used in partitioning or the like of the existing system, by analyzing the existing system.

A system analysis apparatus for analyzing a system containing one or a plurality of programs, which is a first aspect of the present invention, includes means for examining a data item access state in the program, and an analyzer for analyzing association relationships between processes and data items based on the data item access state. Each of the processes is at least one of a program, a set of programs and a program section. The data items are, for example, data items within data files, data items within telegraphic messages, or data items within subprogram interfaces. In this way, by analyzing association relationships between data items and processes as a system analysis, it is possible to acquire information useful in system partitioning etc. More specifically, it is possible to clarify groups of processes of various units (for example, program sections, programs themselves, JOBs, subsystems, etc.), groups of data items, and association relationships between data items and processes.

The above mentioned analyzer may be configured to includes means for quantifying the types of accesses (for example, reference, update, etc.) to the data items and the number of accesses, which are included in the data item access state, and generating the quantified data item access state data. It is possible to display the association degrees between data items and processes as a result of the quantification.

The above mentioned analyzer may also be configured to further include means for correcting the quantified data item access state data according to external requirements including system design requirements. This is in order to designate portions that are to be associated according to the external requirements such as system design requirements.

The above mentioned analyzer may also be configured to further include means for collecting processes which access to a data item satisfying a predetermined condition in the quantified data item access state data.

Still further, the above mentioned analyzer may also configured to further include means for presenting at least one of a partitioning pattern of the data items and a division pattern of the processes, using the quantified data item access state data and collected process information.

It is also possible that the system analysis apparatus of the first aspect of the present invention further includes means for presenting a process interface in the presented process division pattern. This is useful in the design of interfaces at the time of system partitioning.

The system analysis method, which is the second aspect of the present invention, includes the steps of: examining a data item access state in the program, and analyzing association relationships between processes and data items based on the data item access state, wherein each of the processes is at least one of a program, a set of programs and a program section. Variations stated above for the first aspect of the present invention can be applied to the second aspect of the present invention.

This method may be provided as a program that is executed by a computer. In this case, the program is stored in a storage medium or a storage device, such as a floppy disk, CD-ROM, magneto-optical disk, semiconductor memory or hard disk.

BRIEF DESCRIPTION OF THE DRAWINGS

Fig. 1 is a functional block diagram of the present invention;

Fig. 2 is an overall process flowchart of the present invention;

Fig. 3 is a flow chart of an instruction tabulating process of the present invention;

Fig. 4 is a drawing showing an example of data recorded during instruction tabulation;

Fig. 5 is a drawing showing a data item access state table at the first stage;

Fig. 6 is a drawing showing an example of an association coefficient table of the present invention;

Fig. 7 is a drawing showing a data item access state table at the second stage;

Fig. 8 is a drawing showing a data item access state table at the third stage;

Fig. 9 is a flowchart of process accumulation processing of the present invention;

Fig. 10 is a drawing showing a data item access state table at the fourth stage;

Fig. 11 is a flowchart of process and data item division processing of the present invention;

Fig. 12 shows an example of data recorded during process and data item division processing;

Fig. 13 shows an example of data recorded during process and data item division processing;

Fig. 14 is a drawing showing a data item access state table at the fifth stage;

Fig. 15 shows an example of a process interface display of the present invention.

DETAILED DESCRIPATION OF THE PREFERRED EMBODIMENTS

A functional block diagram of the present invention is shown in Fig. 1. Inputs are a program 101, JCL (Job Control Language) 103 and subsystem information 105 used in an existing system. A data item access state tabulation module 107 uses these inputs. Output of the data item access state tabulation module 107 is input to a data item association degree calculating module 109. The data item association degree calculating module 109 performs computation using specially prepared association coefficients. The association coefficients are stored in an association coefficient storage 111. The result of the computation by the data item association degree calculating module 109 is output to a data item association degree correcting module 113. The data item association degree correcting module 113 outputs a result to a process accumulating module 115 after carrying out correction processing. The process accumulating module 115 outputs a result of accumulation processing to a process and data dividing module 117. A processing result of the process and data dividing module 117 is output to an interface display module 119. A processing result of the interface display module 119 is displayed as interface information, and stored in a storage device. The processing result from each module is also stored in the storage device.

The data item access state tabulation module 107 forms the number of accesses, type of access and at what stage the access was made, to each data item, into a table, using (a) the program 101 which is a source program, (b) the JCL 103 which defines names of the programs to be executed, control parameters, and data sets allocated to the programs etc., and (c) the subsystem information 105 which includes subsystem names and information about JOB dependent on the subsystem. The data item association degree calculating module 109 quantifies the data item access state using access type (for example reference, update) and the number of accesses. Since there is an association degree strength for each access type, the data item association degree calculating module 109 calculates a production of an association coefficient corresponding to the access type and the number of accesses by referring to the association coefficient storage 111 which stores the association degree strength (association coefficient) defined for each access type in advance.

The data item association degree correcting module 113 corrects the association degrees related to portions that are to be strongly associated, using input of external requirements such as system design requirements. For example, in the case where the data amount of a particular data item is large compared to others, processing is carried out to cause the association degree for that data item to be increased, or the like. The process accumulating module 115 collects processes that access to data items satisfying a predetermined condition, and gathers the processes together in the data item access state data. By gathering the processes together, it becomes easy to understand the group of the data items and processes. Therefore the process and data dividing module 117 carries out process division and data item partition. Since the interface between the processes is made clear as a result of this process division, the interface display module 119 displays the interface.

The overall process flow of the present invention is shown in Fig. 2, and a detailed description of each step will now be given. First of all, the data item access state tabulation module 107 analyzes the program 101, JCL 103 and subsystem information 105, and tabulates data item access instructions (step 13). An example of this process flow is shown in Fig. 3.

First of all, word disassembly is carried out for a source program written in a programming language, such as COBOL (step 33 in Fig. 3). That is, the source program is disassembled into each instruction. Then, referring to an access instruction table, access states for each data item are stored (step 35). The access instruction table lists instruction groups to be handled as access instructions, such as data referencing instructions (for example READ, GET), data creating instructions (for example WRITE, STORE), data update instructions (for example, REWRITE, MODIFY), data delete instructions (for example DELETE), and data redefining instructions (for example REDEFINE). That is, each instruction enumerated in the access instruction table is recorded, as shown in Fig. 4, for example. Specifically, a program name, section names in that program, step names in those sections, data items accessed and the types of those accesses, are recorded in a storage device. In the example of Fig. 4, section Sec111 of a program called Pgm11 is currently being analyzed, and in step 1 there are reference accesses to data items D11 and D12, and in step 4 there are update accesses to data items D11, D12 and D22. Fig. 4 shows that analysis is in progress, and after the state shown in Fig. 4, the access state for all programs included in the system to be analyzed is recorded.

In step 35, if the access state is recorded for each access, the JCL 103 and the subsystem information 105 are also referenced, and accesses of the same type in each step are counted and a data item access state table is generated (step 37). The data item access state table is a table as shown, for example Fig. 5, and is stored in a storage device. Fig. 5 is a data item access state table for a system called System A. System A contains jobs Job1, Job2, and Job3. Further, Job1 contains programs Pgm11 and Pgm12, Job2 contains programs Pgm21 and Pgm22, and Job3 contains program Pgm31. This information is acquired by referencing the JCL103 and the subsystem information 105. Program Pgm11 contains sections Sec111 and Sec112, program Pgm12 contains section Sec121, program Pgm21 contains section Sec211, program Pgm22 contains section Sec221, and program Pgm31 contains section 311. Further, section Sec111 contains steps 1 to 7, section Sec112 contains steps 8 to 10, section Sec121 contains steps 11 to 13, section Sec211 contains steps 14 to 16, section Sec221 contains steps 17 to 19, and section Sec311 contains steps 20 to 25. The steps are represented by serial numbers for each process unit within a section. This information is recorded in the processing of step 35, as shown in Fig. 4.

On the other hand, file F1 contains records R1 and R2, and file F2 contains record R3. Record R1 includes data items D11 to D14, record R2 includes data items D21 and D22, and record R3 includes data items D31 and D32. This information is acquired from the file structure. If there are reference (for example, READ) accesses, R is recorded in the table of Fig. 5, U is recorded if there are update (for example, UPDATE) accesses, and C is recorded if there are create accesses. Although not used in Fig. 5, D is recorded if there are delete accesses, and A is recorded if there are redefine accesses. The number of accesses is recorded after the letter representing the type of access. R5 indicates that there are five reference accesses. If a small letter a is recorded after the above described alphabetical letters, it means all record processing.

After the data item access state table such as shown in Fig. 5 has been generated, the data item association degree calculating module 109 refers to the association coefficient storage 111 and calculates the degree of data item association (Step 15 in Fig. 2). The degree of association has certain strength depending on the access type, and an association coefficient expresses this strength. Generally, the association degree of an updating access is stronger than that of a reference access and the association degree of create access and delete access is stronger than that of updating access. Also, the association degree of a data item used in a CALL parameter for calling a subprogram is stronger than that of the file access, and the association degree of the shared COPY phrase or local variable is further stronger. Accordingly, the following size relationships exist.

For association relationships:

reference < update < create = delete < redefine

For association media:

File < CALL parameter < COPY phrase < local variable

Based on these size relationships, the specific association coefficients such as shown in Fig. 6 are prepared. However, it is also possible to use numerical values that are different from those in the example of Fig. 6. Further, it is also possible to change the above described size relationships according to some conditions.

In step 15, access types and numbers of accesses tabulated in the data item access state table (Fig. 5) are quantified by multiplying association coefficients corresponding to the access types and numbers of accesses. In the example of Fig. 5, since there are only file accesses, other association media are not used in this example. The quantification results are shown in Fig. 7. The quantified data item access state table is stored in the storage device. For example, in the case of U2 in Fig. 5, since it is an update access, the association coefficient is five and the number of accesses is two, so the degree of association becomes 10. Also, in this example, in the case of all record processing, calculation is carried out with the number of accesses being 10. That is, Ra has an association coefficient of 1, and with the number of accesses as 10 the degree of association becomes 10.

Next the data item association degree correcting module 113 corrects the association degrees in the data item access state table (step 17 in Fig. 2). The association degrees between processes and data have a strength depending not only on the type and number of accesses to data, but also according to the amount of data and response time to be cleared in the system requirement. It is necessary to correct the association degrees in the data item access state table acquired in step 15, taking such system design requirements into consideration. For example, with data items having a large data amount, the association degree is increased in accordance with the data amount. In the example of Fig. 7, since the data amount of file F2 is large, the association degrees for data items included in this file F2 are doubled. The data item access states table after performing correction to double the association degrees is shown in Fig. 8.

If system design requirements includes, such as for example, a requirement that particular processes or data must be associated with each other, it is possible to partition processes or data under that type of restriction. However, if processing subsequent to this embodiment is carried out after causing association degrees of individual data items to be increased (or decreased) as in this example, it becomes possible to find association relationships for data items and processes (for example, between data items, between processes, and between data items and processes) that are completely hidden in cases where explicit association is applied. However, it is also possible to carry out process accumulation (step 21) without carrying out this correction (step 17).

Then, it is confirmed whether or not correction is appropriate (step 19). If correction is not appropriate, processing returns to step 17 and correction is carried out again. Whether or not the correction is appropriate depends upon whether or not subsequent processing, for example process accumulation, is possible.

Next, the process accumulating module 115 accumulates processes with a high association degree (step 21). For example, the processing shown in Fig. 9 is carried out. First of all, a data item with the highest association degree is acquired (step 43). In the example of Fig. 8, this is data item D32. Next, programs using a record containing this data item are collected together in program units (step 45). In the case of Fig. 8, the record that contains the data item D32 is record R3, and programs that use record R3 are Pgm11, Pgm22 and Pgm31. Next, the collected programs are arranged in program units in order of those association degrees (step 47). Program Pgm11 has association degree 140, Pgm22 has association degree 30, and Pgm31 has association degree 40, and therefore they are arranged in the order Pgm11, Pgm31, Pgm22.

It is then judged whether or not there are any remaining elements which satisfy a predetermined condition after programs (column entries in the data item access state table) and records (row entries) that are objects to be arranged have been removed from the table (step 49). In the example of Fig. 8, the programs Pgm11, Pgm31 and Pgm22 are removed as the column entries, which means that Pgm12 and Pgm21 remain. Also, record R3 is removed as the row entries, so records R1 and R2 remain. If there are elements in these remaining portions with association degree larger than a specified threshold value, processing returns to step 43. In the example of Fig. 8, the maximum association degree of the remaining portions is 10. Processing is terminated, as this is smaller than the threshold value.

The result of step 21 is shown in Fig. 10. In step 21, program columns in the data item access state table that are to be arranged are all moved. Accordingly, as shown in Fig. 10, the programs become arranged Pgm31 next to Pgm11, and with Pgm22 next to Pgm31. On the other hand, programs other than those that are to be arranged are not subjected to any further arrangement, which means that Pgm12 and Pgm21 are arranged in numerical order, so Pgm21 is after Pgm12. JOB names are rewritten in accordance with the programs that are arranged. Obviously, sections and steps within the programs are also shifted in accordance with arranging and shifting of programs. If this type of arrangement is carried out, it becomes clear that there is a bias towards data items being used by every program. The data item access state table of Fig. 10 is stored in the storage device.

In the example of Fig. 9, programs have been grouped together based on a record containing the data item with the highest association degree, but it is also possible to group programs together based on other units, for example data items. Also, programs have been grouped together in program units, but it is also possible to group them together in section units or JOB units.

Returning to Fig. 2, the process and data dividing module 117 performs process division and data item partition (step 22). One example of this processing will be described in detail using Fig. 11.

First of all, data items having an association degree of 5 or more are grouped together in section units (step 63). If step 63 is carried out, the processing result for Fig. 10 is shown in Fig. 12.

Specifically, data item D11 has an association degree of 5 or more in sections Sec111, Sec112 and Sec311. Data item D12 has an association degree of 5 or more in sections Sec111 and Sec311. Data item D13 has an association degree of 5 or more in section Sec211. Data item D14 has an association degree of 5 or more in section Sec211. Data item D21 has an association degree of 5 or more in section Sec311. Data item D22 has an association degree of 5 or more in sections Sec111 and Sec311. Data item D31 has an association degree of 5 or more in sections Sec112 and Sec311. Data item D32 has an association degree of 5 or more in sections Sec112, Sec311 and Sec221.

Then, data items are grouped together as one if data items are arranged in the same file in sequence and one of the following conditions is satisfied (step 65).

Condition 1: the range of grouped sections is the same.

Condition 2: there are common sections.

For example, processing is carried out sequentially from the upper line in Fig. 12. Because the first and second lines list data items of the same file F1 and there are common sections, the data items are grouped together (range a). The third line is for a data item in the same file F1, but the range of grouped sections is not the same and there is no common section, which means that the lines from the first to the third line can not be grouped together. If the third and fourth lines are compared, both data items in the third and fourth lines are in the same file F1 and the range of grouped sections is the same, so they can be grouped together (range d). Data items from the third to the fifth line are in the same file F1, but the range of grouped sections is not the same and there is no common section, so they can not be grouped together.

If the fifth line and the sixth line are compared, data items of the same file F1 are listed, and a common section exists in the grouped section range, therefore the fifth line and the sixth line are grouped together (range b). There is a file boundary between the sixth line and the seventh line, and so the seventh line and the eighth line are compared. Looking at the seventh line and the eighth line, data items of the same file F2 are listed, and sections Sec112 and Sec311 are shared, therefore the seventh line and the eight line are grouped together (range c). The above described groupings are illustrated in Fig. 13. The information in Fig. 12 and Fig. 13 is stored in the storage device.

The ranges a to d are shown in the data item access state table in Fig. 14. In the processing of Fig. 11, sections are gathered together for each data item, so association relationships between data items become clear. Association relationships between data items and processes are also shown. In the example of Fig. 14, the association degree between data item D11 and data item D12 is strong, and processes which are strongly associated with these data items D11 and D12 are sections Sec111, Sec112 and Sec311, or programs Pgm11 and Pgm31 (range a). Also, the association degree between data item D13 and data item D14 is strong, and processes which are strongly associated with these data items D13 and D14 are section Sec211 or program Pgm21 (range d). The degree association between data item D21 and data item D22 is strong, and processes which are strongly associated with these data items D21 and D22 are sections Sec111, and Sec311, or programs Pgm11 and Pgm31 (range b). The association degree between data item D31 and data item D32 is strong, and processes which are strongly associated with these data items D31 and D32 are sections Sec112, Sec311 and Sec221, or programs Pgm11, Pgm31and Pgm22 (range c). It is also possible to configure the present invention to display this table to a user.

By grouping in this manner, it will be understood that data items D11 and D12, and D12 and D13 in record R1 are separated, and that they can be placed in different records.

In Fig. 11, data items with an association degree of 5 or more are searched for, but this is only one example and it is also possible to change the value according to conditions of the system to be analyzed. In addition, in the above example, data items are grouped together in section units, but it is also possible to group in program units.

It is also possible to divide processes by using the above ranges a to d. As will be understood from reference to Fig. 14, program Pgm31 or section Sec311 is included in range a to range c, but is not included in range d. Accordingly, it will be understood that it is possible to carry out process division to programs Pgm11, Pgm31 and Pgm22 (or sections Sec111, Sec112 Sec311 and Sec221) derived from ranges a to c, and Pgm21 (or Sec211) derived from range d. Program Pgm12 (or Sec121) does not belong to any range. However, Program Pgm 12 (or Sec121) uses data item D13, and data items D13 and D14 are strongly associated with program Pgm21. Therefore, Pgm12 is grouped with Pgm21. In Fig. 14, a group of Pgm11, Pgm31 and Pgm22, and a group of Pgm12 and Pgm21 are shown as division units.

A more detailed description of the processing described in the above paragraph will now be given. (1) Sections having a high association degree are collected together based on data items D11 and D12. These are Sec111, Sec112 and Sec311. (2) Sections having a high association degree are collected together based on data items D21 and D22. These are Sec111 and Sec311. (3) Sections having a high association degree are collected together based on data items D31 and D32. These are Sec112, Sec311 and Sec221. (4) Section Sec311 is shared by all three of these section groups, so sections strongly associated with data items D11 and D12, D21 and D22, and D31 and D32 are collected together. These are sections Sec111, Sec112, Sec311 and Sec221. In program units this becomes Pgm11, Pgm31 and Pgm22. (5) Sections having a high association degree are collected together based on D13 and D14. This is section 211. Portions with an association degree of 1 are only weakly associated, so they are not collected together (sections Sec311 and Sec121). It will be understood that Sec211 has a weak association degree with other data items. (6) Because section Sec121 has no data items with a strong association degree, and there is a reference to data item D13, it is put into the same group as Sec211 that is strongly associated with D13 and D14. In program units, programs Pgm21 and Pgm12 are in the same group. Therefore, the sections are divided into two division units, namely the group containing Sec111, Sec112 Sec311 and the group containing Sec221, and Sec211 and Sec121.

Returning to Fig. 2, after step 22 has been executed, it is judged whether or not division is appropriate (step 23). If division is not appropriate, processing returns to step 17. Whether or not division is appropriate is determined by judging whether the system design requirements are satisfied, for example. Division is also judged to be not appropriate if division was not possible in step 22.

If it is judged in step 23 that partitioning is appropriate, the interface display module 119 displays an interface (step 25). Here, processes divided in step 22 and data used by the processes are displayed. An example of an interface is shown in Fig. 15. In program units, processes that have been divided are Pgm11, Pgm31 and Pgm22, and Pgm12 and Pgm21. With divided processes as objects, the first group is made object Obj-1, and the second group is made object Obj-2. Data items are public data, which is external data used as interfaces to processes other than the associated processes, and private data, which is internal data used only within the associated processes.

Looking at the division units of Fig. 14, data items D11 and D12 are private data in object

Obj-1, but data item D11 is public data in object Obj-2. On the other hand, data items D13 and D14 are private data in object Obj-2, but data item D14 is public data in object Obj-1. Also, D21 and D22, and D31 and D32 are private data in object Obj-1. The partition of data items and division of processes are also shown in the interface display.

In the interface display, the access types such as reference or update, etc. are indicated on behalf of the values of the association degree shown in Fig. 14. In this way, it becomes clear what type of accesses occur for private data and public data. In Fig. 15, public data is for reference only, and it will also be understood that the association degrees for external processes are low. By dividing into objects, it becomes possible to apply object-oriented technology etc.

In this way, the process interface becomes clear after the division, and it is possible to make use of this in new system design. If clear interface is presented in this way, it becomes easy to decide on the specification of objects that are components, and reductions in design errors can be anticipated.

If each of the above described steps in Fig. 2 are programmed, the present invention can be implemented as a software program. This program is stored in, for example, a storage medium or storage device such as a floppy disk, CD-ROM, Magneto-optical disk, semiconductor memory, hard disk, etc. The program is then executed by a computer. A computer for executing the system analysis program of the present invention can operate as a system analyzer.

It is also possible to implement each function shown in Fig. 1 as a special electronic circuit, or as a combination of electronic circuits and programs, and realize the system analyzer of the present invention by using those circuits. The functional blocks shown in Fig. 1 are one example, and it is possible to combine a plurality of functional blocks into one, or similarly to divide a single functional block into a plurality of blocks.

By utilizing the present invention as described above, it is possible to analyze the association degree between processes, it is possible to examine a scheme for reconstructing an existing system in process units, and it is possible to prevent increase in development costs and time scales. Also, by collecting together groups of processes with strong association degrees, groups of subsystems, groups of JOBs inside subsystems, groups of programs inside JOBs, groups of sections inside programs and clusters of instruction groups inside sections can be carried out reliably. In addition, it becomes possible to make programs or portions of programs reusable. Further, it is also possible to partition data and files instead of dividing processes, and it becomes possible to carry out data normalization support and optimum rearrangement of data.

As described above, the present invention could provide a system analysis apparatus and method, and program for analyzing an existing system. It could also provide information to be used in partitioning an existing system by analyzing the existing system.

Although the present invention has been described with respect to a specific preferred embodiment thereof, various change and modifications may be suggested to one skilled in the art, and it is intended that the present invention encompass such changes and modifications as fall within the scope of the appended claims.

What is claimed is:

1. A system analysis apparatus for analyzing a system containing one or a plurality of programs,

comprising:

means for examining a data item access state in said program; and

an analyzer for analyzing association relationships between processes and data items based on said data item access state, each said process being at least one of a program, a set of programs and a program section.

2. The system analysis apparatus of claim 1, wherein said analyzer comprises:

means for quantifying the types of accesses to the data items and the number of accesses, which are included in said data item access state, and generating the quantified data item access state data.

3. The system analysis apparatus of claim 2, wherein said analyzer further comprises:

means for correcting said quantified data item access state data according to an external requirement including a system design requirement.

4. The system analysis apparatus of claim 3, wherein said analyzer further comprises:

means for collecting processes that access to a data item satisfying a predetermined condition in said quantified data item access state data.

5. The system analysis apparatus of claim 4, wherein said analyzer further comprises:

means for presenting at least one of a partitioning pattern of the data items and a division pattern of the processes, using the quantified data item access state data and the collected process information.

6. The system analysis apparatus of claim 5, further comprising:

means for presenting a process interface in the presented division pattern of the processes.

7. A system analyzing method, for analyzing a system containing one or a plurality of programs, comprising the steps of:

examining a data item access state in the program; and analyzing association relationships between processes and data items based on said data item access state, each said process being at least one of a program, a set of programs and a program section.

8. The system analysis method of claim 7, wherein said analyzing step comprises a step of:

quantifying the types of accesses to the data items and the number of accesses, which are included in said data item access state, and generating the quantified data item access state data.

9. The system analysis method of claim 8, wherein said analyzing step further comprises a step of:

correcting said quantified data item access state data according to an external requirement including a system design requirement.

10. The system analysis method of claim 9, wherein said analyzing step further comprises a step of:

collecting processes that access to a data item satisfying a predetermined condition in said quantified data item access state data.

11. The system analysis method of claim 10, wherein said analyzing step further comprises a step of:

presenting at least one of a partitioning pattern of the data items and a division pattern of the processes, using the quantified data item access state data and the collected process information.

12. The system analysis method of claim 11, further comprising a step of:

presenting a process interface in the presented division pattern of the processes.

13. A storage medium for storing an analysis program for analyzing a system containing one or a plurality of programs, said analysis program causing a computer to execute the steps of:

examining a data item access state in the program; and

analyzing association relationships between processes and data items based on said data item access state, each said process being at least one of a program, a set of programs and a program section.

14. The storage medium of claim 13, wherein said analyzing step comprises a step of:

quantifying the types of accesses to the data items and the number of accesses, which are included in said data item access state, and generating the quantified data item access state data.

15. The storage medium of claim 14, wherein said analyzing step further comprises a step of:

correcting said quantified data item access state data according to an external requirement including a system design requirement.

16. The storage medium of claim 15, wherein said analyzing step further comprises a step of:

collecting processes that access to a data item satisfying a predetermined condition in said quantified data item access state data.

17. The storage medium of claim 16, wherein said analyzing step further comprises a step of:

presenting at least one of a partitioning pattern of the data items and a division pattern of the processes, using the quantified data item access state data and the collected process information.

18. The storage medium of claim 11, wherein said analysis program causes said computer to further execute a step of:

presenting a process interface in the presented division pattern of the processes.

Abstract

The object of the present invention is to analyze association relationships between data items and processes being used in an existing system. By analyzing a program or the like, the type and numbers of access instructions to data items in each step of the program is examined (step 13). Then, data item association degree is calculated from the type of access instruction etc., and the values of the association degree are corrected using system requirements etc (step 15 and step 17). Processes having high association degrees are collected together by referencing the corrected values of the association degree (step 21). Based on the groups of collected processes, process division and data item partitioning is performed (step 23). Interfaces between processes are then displayed based on divided processes and partitioned data items (step 25).

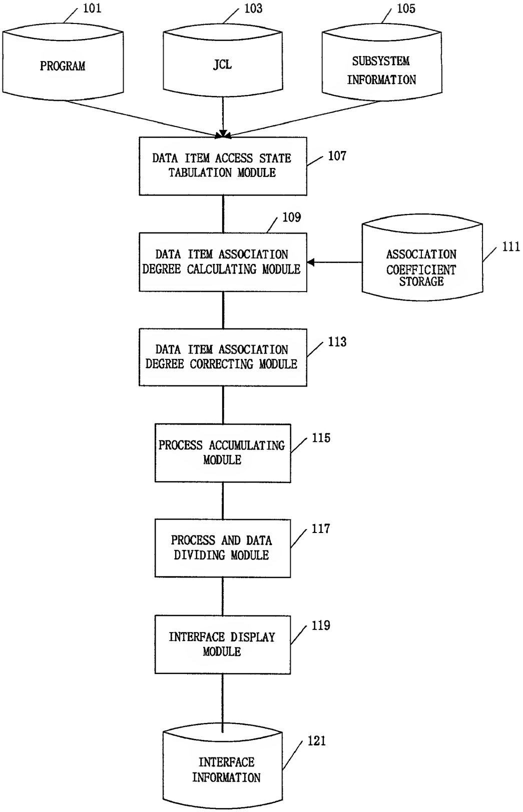

101
PROGRAM
103
JCL
105
SUBSYSTEM INFORMATION
DATA ITEM ACCESS STATE TABULATION MODULE
107
109
DATA ITEM ASSOCIATION DEGREE CALCULATING MODULE
ASSOCIATION COEFFICIENT STORAGE
111
DATA ITEM ASSOCIATION DEGREE CORRECTING MODULE
113
PROCESS ACCUMULATING MODULE
115
PROCESS AND DATA DIVIDING MODULE
117
INTERFACE DISPLAY MODULE
119
INTERFACE INFORMATION
121

FIG.1

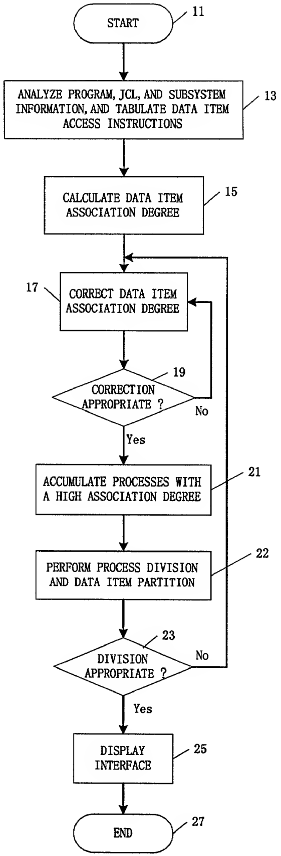

START
11
ANALYZE PROGRAM, JCL, AND SUBSYSTEM INFORMATION, AND TABULATE DATA ITEM ACCESS INSTRUCTIONS
13
CALCULATE DATA ITEM ASSOCIATION DEGREE
15
17
CORRECT DATA ITEM ASSOCIATION DEGREE
19
CORRECTION APPROPRIATE ?
No
Yes
ACCUMULATE PROCESSES WITH A HIGH ASSOCIATION DEGREE
21
PERFORM PROCESS DIVISION AND DATA ITEM PARTITION
22
23
DIVISION APPROPRIATE ?
No
Yes
DISPLAY INTERFACE
25
END
27

FIG.2

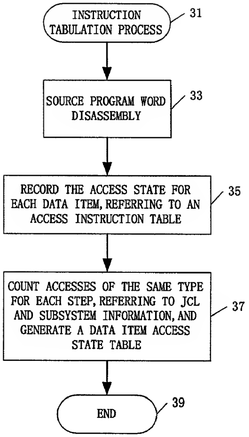

INSTRUCTION TABULATION PROCESS
31
SOURCE PROGRAM WORD DISASSEMBLY
33
RECORD THE ACCESS STATE FOR EACH DATA ITEM, REFERRING TO AN ACCESS INSTRUCTION TABLE
35
COUNT ACCESSES OF THE SAME TYPE FOR EACH STEP, REFERRING TO JCL AND SUBSYSTEM INFORMATION, AND GENERATE A DATA ITEM ACCESS STATE TABLE
37
END
39

FIG.3

PROGRAM	SECTION	STEP	DATA ITEM	ACCESS TYPE
Pgm11	Sec111	1	D11	READ
Pgm11	Sec111	1	D12	READ
Pgm11	Sec111	4	D11	UPDATE
Pgm11	Sec111	4	D12	UPDATE
Pgm11	Sec111	4	D22	UPDATE
·	·		·	·
·	·		·	·
·	·		·	·

FIG.4

SYSTEM			SystemA																								
JOB			Job1													Job2						Job3					
PROGRAM			Pgm11										Pgm12			Pgm21			Pgm22			Pgm31					
SECTION			Sec111							Sec112			Sec121			Sec211			Sec221			Sec311					
STEP#			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
FILE	REC	DATA																									
F1	R1	D11	R1			U1				R5						R1			R1			U1					
		D12	R1			U1																U1					
		D13											R1				U2										
		D14																U1				R1					
	R2	D21																					C1				
		D22				U1																	C1				
F2	R3	D31								Ra														C1			
		D32								Ra	Ua								Ra					C1			

FIG.5

		ASSOCIATION RELATIONSHIP				
		REFERENCE	UPDATE	CREATE	DELETE	REDEFINE
ASSOCIATION MEDIUM	FILE	1	5	10	10	
	CALL PARAMETER	2	7	12	12	
	COPY PHRASE	3	9			20
	LOCAL VARIABLE	50	60			90

FIG.6

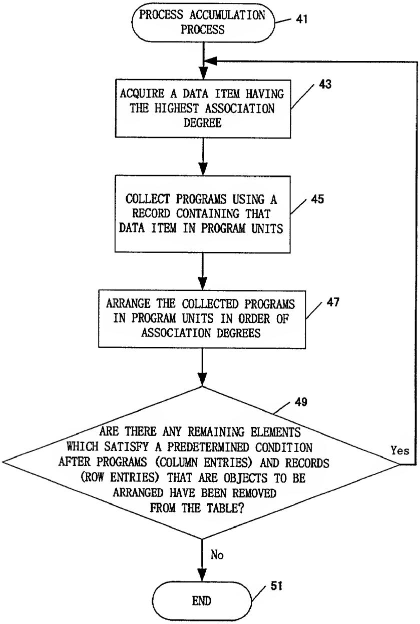

PROCESS ACCUMULATION PROCESS
41
ACQUIRE A DATA ITEM HAVING THE HIGHEST ASSOCIATION DEGREE
43
COLLECT PROGRAMS USING A RECORD CONTAINING THAT DATA ITEM IN PROGRAM UNITS
45
ARRANGE THE COLLECTED PROGRAMS IN PROGRAM UNITS IN ORDER OF ASSOCIATION DEGREES
47
49
ARE THERE ANY REMAINING ELEMENTS WHICH SATISFY A PREDETERMINED CONDITION AFTER PROGRAMS (COLUMN ENTRIES) AND RECORDS (ROW ENTRIES) THAT ARE OBJECTS TO BE ARRANGED HAVE BEEN REMOVED FROM THE TABLE?
Yes
No
END
51

FIG.9

SYSTEM			SystemA																								
JOB			Job1													Job2						Job3					
PROGRAM			Pgm11										Pgm12			Pgm21			Pgm22			Pgm31					
SECTION			Sec111							Sec112			Sec121			Sec211			Sec221			Sec311					
STEPs			1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
FILE	REC	DATA																									
F1	R1	D11	1			5				5						1			1			5					
		D12	1			5																5					
		D13											1				10										
		D14																5				1					
	R2	D21																					10				
		D22				5																	10				
F2	R3	D31								10														10			
		D32								10	50								10					10			

FIG.7

		SYSTEM	SystemA																								
		JOB	Job1													Job2						Job3					
		PROGRAM	Pgm11										Pgm12			Pgm21			Pgm22			Pgm31					
		SECTION	Sec111							Sec112			Sec121			Sec211			Sec221			Sec311					
		STEPs	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
FILE	REC	DATA																									
F1	R1	D11	1			5				5						1			1			5					
		D12	1			5																5					
		D13											1				10										
		D14																5				1					
	R2	D21																					10				
		D22				5																	10				
F2	R3	D31								20														20			
		D32								20	100								20					20			

FIG.8

	SYSTEM		SystemA																							
	JOB		Job1										Job3						Job2			Job1			Job2		
	PROGRAM		Pgm11										Pgm31						Pgm22			Pgm12			Pgm21		
	SECTION		Sec111							Sec112			Sec311						Sec221			Sec121			Sec211		
	STEPs		1	2	3	4	5	6	7	8	9	10	20	21	22	23	24	25	17	18	19	11	12	13	14	15	16
FILE	REC	DATA																									
F1	R1	D11	1			5				5			5						1						1		
		D12	1			5							5														
		D13																				1				10	
		D14											1														5
	R2	D21												10													
		D22				5								10													
F2	R3	D31								20					20												
		D32								20	100				20				20								

FIG.10

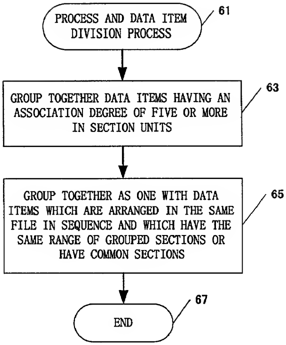

PROCESS AND DATA ITEM DIVISION PROCESS
61
GROUP TOGETHER DATA ITEMS HAVING AN ASSOCIATION DEGREE OF FIVE OR MORE IN SECTION UNITS
63
GROUP TOGETHER AS ONE WITH DATA ITEMS WHICH ARE ARRANGED IN THE SAME FILE IN SEQUENCE AND WHICH HAVE THE SAME RANGE OF GROUPED SECTIONS OR HAVE COMMON SECTIONS
65
END
67

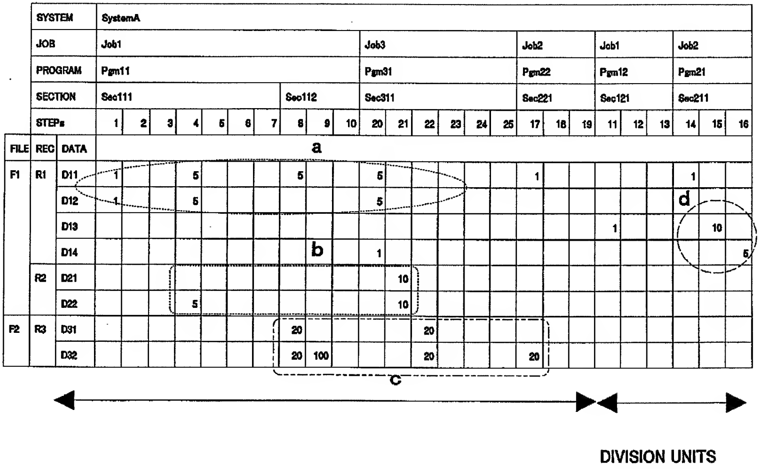
FIG.11

FILE NAME	DATA ITEM NAME	SECTION NAME
F1	D11	Sec111,Sec112,Sec311
F1	D12	Sec111,Sec311
F1	D13	Sec211
F1	D14	Sec211
F1	D21	Sec311
F1	D22	Sec111,Sec311
F2	D31	Sec112,Sec311
F2	D32	Sec112,Sec311,Sec221

FIG.12

FILE NAME	DATA ITEM NAME	SECTION NAME	
F1	D11	Sec111,Sec112,Sec311	a
F1	D12	Sec111,Sec311	
F1	D13	Sec211	d
F1	D14	Sec211	
F1	D21	Sec311	b
F1	D22	Sec111,Sec311	
F2	D31	Sec112,Sec311	c
F2	D32	Sec112,Sec311,Sec221	

FIG.13



	SYSTEM		SystemA																								
	JOB		Job1										Job3						Job2			Job1			Job2		
	PROGRAM		Pgm11										Pgm31						Pgm22			Pgm12			Pgm21		
	SECTION		Sec111							Sec112			Sec311						Sec221			Sec121			Sec211		
	STEPs		1	2	3	4	5	6	7	8	9	10	20	21	22	23	24	25	17	18	19	11	12	13	14	15	16
FILE	REC	DATA																									
F1	R1	D11	1			5				5			5						1						1		
		D12	1			5							5														
		D13																				1				10	
		D14											1														5
	R2	D21												10													
		D22				5								10													
F2	R3	D31								20					20												
		D32								20	100				20				20								

FIG.14

	OBJECT			
	obj-1(pgm11,pgm31,pgm22)		obj-2(pgm12,pgm21)	
	PRIVATE DATA	PUBLIC DATA	PUBLIC DATA	PRIVATE DATA
D11	(REFERENCE,UPDATE)		(REFERENCE)	
D12	(REFERENCE,UPDATE)			
D13				(REFERENCE,UPDATE)
D14		(REFERENCE)		(UPDATE)
D21	(CREATE)			
D22	(CREATE,UPDATE)			
D31	(CREATE,REFERENCE)			
D32	(CREATE,REFERENCE,UPDATE)			

FIG.15

99-15241

PTO/SB/106 (8-96)
Approved for use through 9/30/98. OMB 0651-0032
Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

Declaration and Power of Attorney For Patent Application

特許出願宣言書及び委任状

Japanese Language Declaration

日本語宣言書

下記の氏名の発明者として、私は以下の通り宣言します。

As a below named inventor, I hereby declare that:

私の住所、私書箱、国籍は下記の私の氏名の後に記載された通りです。

My residence, post office address and citizenship are as stated next to my name.

下記の名称の発明に関して請求範囲に記載され、特許出願している発明内容について、私が最初かつ唯一の発明者（下記の氏名が一つの場合）もしくは最初かつ共同発明者であると（下記の名称が複数の場合）信じています。

____

____

I believe I am the original, first and sole inventor (if only one name is listed below) or an original, first and joint inventor (if plural names are listed below) of the subject matter which is claimed and for which a patent is sought on the invention entitled

System Analysis Apparatus and Method, and Strage Medium Storing System Analysis Program

上記発明の明細書（下記の欄でx印がついていない場合は、本書に添付）は、

the specification of which is attached hereto unless the following box is checked:

□ __月__日に提出され、米国出願番号または特許協定条約国際出願番号を__________とし、
（該当する場合）__________ に訂正されました。

□ was filed on ______________________
as United States Application Number or
PCT International Application Number
______________ and was amended on
______________ (if applicable).

私は、特許請求範囲を含む上記訂正後の明細書を検討し、内容を理解していることをここに表明します。

I hereby state that I have reviewed and understand the contents of the above identified specification, including the claims, as amended by any amendment referred to above.

私は、連邦規則法典第３７編第１条５６項に定義されるとおり、特許資格の有無について重要な情報を開示する義務があることを認めます。

I acknowledge the duty to disclose information which is material to patentability as defined in Title 37, Code of Federal Regulations, Section 1.56.

Burden Hour Statement: This form is estimated to take 0.4 hours to complete. Time will vary depending upon the needs of the individual case. Any comments on the amount of time you are required to complete this form should be sent to the Chief Information Officer, Patent and Trademark Office, Washington, DC 20231. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: Commissioner of Patents and Trademarks, Washington, DC 20231.

Japanese Language Declaration
（日本語宣言書）

私は、米国法典第３５編１１９条(a)-(d)項又は３６５条(b)項に基き下記の、米国以外の国の少なくとも一ヵ国を指定している特許協力条約３６５(a)項に基ずく国際出願、又は外国での特許出願もしくは発明者証の出願についての外国優先権をここに主張するとともに、優先権を主張している、本出願の前に出願された特許または発明者証の外国出願を以下に、枠内をマークすることで、示しています。

I hereby claim foreign priority under Title 35, United States Code, Section 119 (a)-(d) or 365(b) of any foreign application(s) for patent or inventor's certificate, or 365(a) of any PCT international application which designated at least one country other than the United States, listed below and have also identified below, by checking the box, any foreign application for patent or inventor's certificate, or PCT international application having a filing date before that of the application on which priority is claimed.

Prior Foreign Application(s)
外国での先行出願

			Priority Not Claimed 優先権主張なし
11-271608	Japan	September 27, 1999	□
(Number) (番号)	(Country) (国名)	(Day/Month/Year Filed) (出願年月日)	
			□
(Number) (番号)	(Country) (国名)	(Day/Month/Year Filed) (出願年月日)	

私.ゝ、第３５編米国法典１１９条(e)項に基いて下記の米国特許出願規定に記載された権利をここに主張いたします。

I hereby claim the benefit under Title 35, United States Code, Section 119(e) of any United States provisional application(s) listed below.

(Application No.) (出願番号)	(Filing Date) (出願日)	(Application No.) (出願番号)	(Filing Date) (出願日)

私は、下記の米国法典第３５編１２０条に基いて下記の米国特許出願に記載された権利、又は米国を指定している特許協力条約３６５条(c)に基ずく権利をここに主張します。また、本出願の各請求範囲の内容が米国法典第３５編１１２条第１項又は特許協力条約で規定された方法で先行する米国特許出願に開示されていない限り、その先行米国出願書提出日以降で本出願書の日本国内または特許協力条約国際提出日までの期間中に入手された、連邦規則法典第３７編１条５６項で定義された特許資格の有無に関する重要な情報について開示義務があることを認識しています。

I hereby claim the benefit under Title 35, United States Code, Section 120 of any United States application(s), or 365(c) of any PCT international application designating the United States, listed below and, insofar as the subject matter of each of the claims of this application is not disclosed in the prior United States or PCT international application in the manner provided by the first paragraph of Title 35, United States Code Section 112, I acknowledge the duty to disclose information which is material to patentability as defined in Title 37, Code of Federal Regulations, Section 1.56 which became available between the filing date of the prior application and the national or PCT international filing date of application.

(Application No.) (出願番号)	(Filing Date) (出願日)	(Status: Patented, Pending, Abandoned) (現況：特許許可済、係属中、放棄済)
(Application No.) (出願番号)	(Filing Date) (出願日)	(Status: Patented, Pending, Abandoned) (現況：特許許可済、係属中、放棄済)

私は、私自身の知識に基ずいて本宣言書中で私が行なう表明が真実であり、かつ私の入手した情報と私の信じるところに基ずく表明が全て真実であると信じていること、さらに故意になされた虚偽の表明及びそれと同等の行為は米国法典第１８編第１００１条に基ずき、罰金または拘禁、もしくはその両方により処罰されること、そしてそのような故意による虚偽の声明を行なえば、出願した、又は既に許可された特許の有効性が失われることを認識し、よってここに上記のごとく宣誓を致します。

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

PTO/SB/106 (8-96)
Approved for use through 9/30/98. OMB 0651-0032
Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

Japanese Language Declaration
(日本語宣言書)

委任状：　私は下記の発明者として、本出願に関する一切の手続きを米特許商標局に対して遂行する弁理士または代理人として、下記の者を指名いたします。（弁護上、または代理人の氏名及び登録番号を明記のこと）

POWER OF ATTORNEY: As a named inventor, I hereby appoint the following attorney(s) and/or agent(s) to prosecute this application and transact all business in the Patent and Trademark Office connected therewith ***(list name and registration number)***

James D Halsey, Jr., 22,729; Harry John Staas, 22,010, David M. Pitcher, 25,908; John C. Garvey, 28,607; J. Randall Beckers, 30,358; William F. Herbert, 31,024; Richard A. Gollhofer, 31,106, Mark J. Henry, 36,162; Gene M. Garner II, 34,172; Michael D Stein, 37,240; Paul I. Kravetz, 35,230, Gerald P. Joyce, III, 37,648; Todd E. Marlette, 35,269; Harlan B. Williams, Jr., 34,756; George N Stevens, 36,938, Michael C. Soldner, 41,455, Norman L. Ourada, 41,235; Kevin R. Spivak, P-43,148; and William M. Schertler, 35,348 (agent)

書類送付先

Send Correspondence to:

STAAS & HALSEY
700 Eleventh Street, N.W.
Suite 500
Washington, D.C. 20001

直接電話連絡先：　（名前及び電話番号）

Direct Telephone Calls to: ***(name and telephone number)***

STAAS & HALSEY
(202) 434-1500

唯一または第一発明者名	**Full name of sole or first inventor** Jyunichi KAMAKURA	
発明者の署名　　日付	**Inventor's signature** Kamakura Jyunichi	**Date** April 10, 2000
住所	**Residence** Chiba-shi, Chiba Japan	
国籍	**Citizenship** Japan	
私書箱	**Post Office Address** c/o FUJITSU LIMITED, 1-1, Kamikodanaka 4-chome, Nakahara-ku,	
	Kawasaki-shi, Kanagawa 211-8588 Japan	
第二共同発明者	**Full name of second joint inventor, if any** Kazuya BABA	
第二共同発明者　　日付	**Second inventor's signature** Kazuya Baba	**Date** April 10, 2000
住所	**Residence** Chiba-shi, Chiba Japan	
国籍	**Citizenship** Japan	
私書箱	**Post Office Address** c/o FUJITSU LIMITED, 1-1, Kamikodanaka 4-chome, Nakahara-ku,	
	Kawasaki-shi, Kanagawa 211-8588 Japan	

（第三以降の共同発明者についても同様に記載し、署名をすること）

(Supply similar information and signature for third and subsequent joint inventors.)